

UNITED STATES PATENT APPLICATION

for

Deterministic Shut Down of Memory Devices

in response to a System Warm Reset

Inventors:

Zohar Bogin

Surya Kareenahalli

Anoop Mukker

David Sastry

Tuong Trieu

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

12400 Wilshire Boulevard

Los Angeles, CA 90025-1030

(408) 720-8300

Attorney's Docket No.: 042390.P17980

"Express Mail" mailing label number: EV336588224US

Date of Deposit: 10/24/03

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Juanita Briscoe

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

## **Deterministic Shut Down of Memory Devices in response to a System Warm Reset**

### **FIELD OF INVENTION**

**[0001]** The present invention relates to computing technology, and more particularly, to deterministically shutting down memory devices in response to a system warm reset.

### **BACKGROUND**

**[0002]** In general, there are two types of reset in a computer system, namely, a cold reset and a warm reset. To cold reset a computer system is to power down substantially all or all of the computer system and then to restart the computer system. In contrast, to warm reset the computer system is to restart the computer system, which is already turned on, via the operating system of the computer system. During the warm reset, the power in the computer system remains substantially stable. A user or a software application running on the computer system may initiate the warm reset.

**[0003]** Currently, many memory devices do not support an input control that permits an asynchronous logic reset to put the memory devices into a known initial logic state. To reach a known initial logic state for proper operation in the memory devices, a power cycling sequence is performed to initialize the memory devices. In other words, the memory devices go through a sequence of powering off and then on in order to be properly initialized to a known logic state, which is also referred to as a known state.

**[0004]** While running a power cycling sequence in the memory devices does not significantly impact the computer system when the computer system is powered up from a cold state, the power cycling sequence should not be run when the computer system is to be reset from a warm state because by definition a power cycling sequence is not a

warm reset. As a result of no power cycling sequence when coming out of a warm reset, the memory devices will start with whatever state the memory devices were in previously before the warm reset. If the previous state is a known initial state, then the warm reset does not create a problem. However, if the previous state is an arbitrary unknown state, then the system cannot boot up because the memory devices are in the arbitrary unknown memory state after the warm reset.

**[0005]** In a prior art memory controller, also commonly referred to as a memory controller hub (MCH), the MCH is reset asynchronously in response to a system warm reset. The asynchronous reset of the MCH results in a non-deterministic timing relationship of interfacing signals (e.g., RAS#, CAS#, WE#, CKE controls, and CK/CKB clocks) between the MCH and the memory devices. This non-deterministic timing relationship may violate the setup and hold timing requirements of the interfacing signals relative to the clock signals of the memory devices. Furthermore, such a violation may cause the memory devices to enter into one or more arbitrary unknown states. To recover from the unknown states, the memory devices have to go through a power cycling sequence after the asynchronous reset.

**[0006]** However, it is problematic to force the memory devices to go through a power cycling sequence as this action by definition is a cold reset, as opposed to the intended warm reset.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present invention will be understood more fully from the detailed description that follows and from the accompanying drawings, which however, should not be taken to limit the appended claims to the specific embodiments shown, but are for explanation and understanding only.

[0008] Figure 1 shows a flow diagram of one embodiment of a process to deterministically shut down memory devices.

[0009] Figure 2 shows a state diagram of one embodiment of a state machine of a memory controller.

[0010] Figure 3 shows one embodiment of a memory controller.

[0011] Figure 4 shows another embodiment of a memory controller.

[0012] Figure 5 shows an exemplary embodiment of a computer system.

## DETAILED DESCRIPTION

[0013] In the following description, numerous specific details are set forth. However, it is understood that some embodiments may be practiced without these specific details. In other instances, well-known circuits, structures, and techniques have not been shown in detail in order not to obscure the understanding of this description. Furthermore, references to “one embodiment” in the following description may or may not be directed to the same embodiment.

[0014] Figure 1 shows a flow diagram of one embodiment of a process to shut down memory devices in a system in response to a warm reset in the system. The process is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine), or a combination of both. Referring to Figure 1, processing logic checks whether the memory devices have been initialized (processing block 110). In one embodiment, processing logic reads a memory initialization register bit, which is not affected by the warm reset. In one embodiment, the bit is set to “1” when the memory devices are not initialized and “0” when the memory devices are initialized. If the memory devices have not been initialized, processing logic checks whether there is a warm reset in the system (processing block 112). The warm reset in the system may also be referred to as a main reset or a system main reset. If there is a warm reset in the system, processing logic forces a cold reset in the memory devices (processing block 114). In one embodiment, processing logic causes the power supplies to the memory devices to be turned off and on, thus subjecting the memory devices through a power cycling sequence in order to put the memory devices into a known logic state. If there is

no warm reset in the system, processing logic checks whether the memory devices have been initialized again by repeating the processing block 110.

**[0015]** When the memory devices have been initialized, processing logic enables a deterministic shutdown mode (processing block 116). In one embodiment, processing logic further updates the memory initialization register bit to indicate that the memory devices have been initialized. While in the deterministic shutdown mode, processing logic checks whether there is a warm reset in the system (processing block 118). If there is no warm reset in the system, processing logic continues monitoring for a warm reset in the system. Otherwise, processing logic gates a main reset signal indicating a warm reset in the system (processing block 120). Gating the main reset signal prevents the memory controller from being forced into a reset, which may result in an indeterminate relationship of command interfacing signals to the memory devices causing the memory devices to enter into an arbitrary unknown state not recoverable unless the memory devices go through a power cycling sequence.

**[0016]** To avoid putting the memory devices into an arbitrary unknown state, processing logic performs a series of operations to ensure the memory devices are put into a known state. The known state is a state in which the clocks of the memory devices are disabled such that the memory devices are insensitive to other input control signals provided to the memory devices. The memory devices can recover correctly from the known state without powering down. An example of the known state is a self-refresh state. In one embodiment, processing logic prevents pending transactions in one or more memory request queues from being serviced while allowing transactions already committed to finish before the memory devices have entered into the known state

(processing block 130). Processing logic may signal a memory request arbiter in a memory controller to block service to the pending transactions. Furthermore, processing logic may reset one or more logic blocks that are no longer used to put the memory devices into a known state before the memory devices have entered into the known state (processing block 132).

**[0017]** When the committed transactions are finished, processing logic puts the memory devices into a known state (processing block 135). When the memory devices are in a known state, processing logic disables the memory devices (processing block 140). Processing logic then passes the main reset signal to allow a memory controller to reset (processing block 145). After resetting the memory controller, processing logic may disable a memory clock (processing logic 150). Then processing logic completes the warm reset of the memory controller (processing block 160). Processing logic re-enables the memory clock after completing the warm reset.

**[0018]** The operations described above to shut down memory devices in response to a system warm reset may be implemented using, for instance, a state machine in a memory controller and a software routine. Furthermore, the state machine may be implemented with application-specific circuitry, general-purpose circuitry, or a combination of both. Figure 2 shows a state diagram of one embodiment of a state machine of a memory controller. Initially, the state machine is in an idle state 210 before the memory devices have been initialized. The state machine remains in the idle state 210 as a deterministic shutdown mode has not been enabled. If a system main reset, also known as a warm reset, is detected while the state machine is in the idle state 210, the state machine may pass a signal indicating the warm reset. Since the memory devices

have not yet been initialized, the memory devices cannot respond to commands such as entering self-refresh. Hence, the memory devices are forced into a cold reset in response to the warm reset being initiated in the system.

**[0019]** Upon the computer system restart after the warm reset, the software may then cause the memory devices to go into a cold reset by running a power supply cycling sequence. In one embodiment, a basic input/output system (BIOS) program causes the power supplies to the memory devices to be turned off and on, thus subjecting the memory devices through a power cycling sequence to put the memory devices into the cold reset.

**[0020]** After the memory devices have been initialized, the deterministic shutdown mode is enabled to allow true handling of the warm reset. When the deterministic shutdown mode is enabled, the state machine goes into an armed state 220. During the armed state 220, the signal indicating that the system main reset has started is gated from the memory controller and the memory devices such that selective logic blocks of the memory controller may not respond to the signal initially. For example, logic components of the memory controller that are associated with the state machine, such as counters, timers, and flags, are gated from the main system reset. The state machine remains in the armed state 220 until the main system reset is detected by monitoring the signal.

**[0021]** In response to the signal, the state machine goes into the state 230. The finishing cycle of the memory controller is in progress during the state 230. In one embodiment, pending transactions in a queue of memory transactions are blocked from the memory devices when the state machine is in the state 230. A transaction already

committed to by the memory controller may be allowed to finish within a predetermined time interval. In one embodiment, the time interval for finishing committed transactions is predetermined and the state machine uses a programmable counter to time the interval. Alternatively, logic blocks performing the committed transactions may send signals to the state machine to inform the state machine of the completion of the committed transactions. Moreover, selective logic components of the memory controller may be reset at various stages during this time interval if the respective logic components are no longer used to complete the reset of the memory devices. The resetting of selective logic components during this time interval is also referred to as a staged reset of the memory controller.

**[0022]** When the committed transactions have been completed, the state machine goes into the state 240 to signal the memory controller to put the memory devices into a known state, for instance, a self-refresh state. Again, selective logic components of the memory controller may be reset at various stages during this time interval when the respective logic components are no longer used to complete the shut down of the memory devices. The resetting of selective logic components during this time interval is also referred to as a staged reset of the memory controller.

**[0023]** Furthermore, the state machine may cause the memory controller to de-assert a clock enable signal in order to disable a memory clock during the state 240. The state machine may wait a predetermined period of time interval for the memory devices to enter into the known state. In one embodiment, the state machine uses a programmable counter to time the interval. When the time interval has elapsed, the memory devices have entered the known state. Alternatively, the state machine may

monitor the memory devices to determine whether the memory devices have entered the known state.

**[0024]** After the memory devices have gone into the known state, the state machine may go into the state 250 to complete the warm reset of the memory controller. In one embodiment, the state machine passes the gated main reset signal to cause the memory controller to completely reset. When the state machine and the associated logic blocks in the memory controller have been asynchronously reset, the state machine returns to the initial idle state 210.

**[0025]** Figure 3 shows one embodiment of a memory controller for controlling memory devices in a system. The memory controller includes an enable switch 310, a gate 320, a detector 330, a sequencer 340, and a memory cycle tracker and command generator 350. Referring to Figure 3, the enable switch 310 enables a deterministic shutdown mode in response to an input or a signal from a basic input/output system (BIOS) program. In one embodiment, the BIOS program activates the enable switch 310 to enable the deterministic shutdown mode once the memory devices have been initialized. The enable switch 310 outputs a signal to the gate 320.

**[0026]** Furthermore, the gate 320 receives a main reset signal 302 from the system, which indicates whether there is a system warm reset. The main reset signal 302 is also input to the detector 330. The detector provides an output to the sequencer 340. The sequencer 340 outputs signals to the gate 320 and the memory cycle tracker and command generator 350. The memory cycle tracker and command generator 350 also receives the output from the gate 320. The output of the memory cycle tracker and

command generator 350 drives the memory devices in response to the signals from the sequencer 340 and/or the gate 320.

**[0027]** When the memory devices are not initialized yet, the enable switch 310 disables the deterministic shutdown mode to allow the gate 320 to pass the main reset signal 302 to the memory cycle tracker and command generator 350 as well as other logic blocks of the memory controller 300. If the main reset signal 302 indicates that a warm reset occurs in the system, the gate 320 passes the main reset signal 302 to cause the memory controller 300 to reset and to allow the BIOS program upon a system restart to force a cold reset in the memory devices. In one embodiment, the cold reset includes power cycling the memory devices.

**[0028]** When the memory devices have been initialized, the enable switch 310 enables the deterministic shutdown mode to cause the gate 320 to block the main reset signal 302 from the memory cycle tracker and command generator 350 and other logic blocks of the memory controller 300. When there is a warm reset, the main reset signal 302 is asserted. In response to the asserted main reset signal 302, the detector 330 sends an output to the sequencer 340 to cause the sequencer 340 to go through a sequence of operations to put the memory devices into a known state.

**[0029]** In one embodiment, the sequencer 340 prevents pending transactions in a memory queue from being serviced while allowing committed transactions to be finished. The sequencer 340 may use a timer 345 to wait a predetermined period of time such that the committed transactions are finished by the end of the period. When the committed transactions are finished, the sequencer 340 sends signals to the memory cycle tracker and command generator 350 to cause the memory cycle tracker and command generator

350 to put the memory devices into a known state, such as, for example, a self-refresh state.

**[0030]** After the memory devices have entered the known state, the sequencer 340 signals the gate 320 to pass the main reset signal 302 to other logic blocks of the memory controller 300. In response to the main reset signal, all logic blocks of the memory controller 300 may reset, i.e., the memory controller 300 may completely reset.

**[0031]** Figure 4 shows an alternate embodiment of the memory controller. The memory controller 400 includes an enable switch 410, a gate 420, a detector 430, a sequencer 440, and a memory cycle tracker and command generator 450. The enable switch 410, the gate 420, and the detector 430 function in a substantially similar way as the corresponding components shown in Figure 3. However, unlike the sequencer 340, which uses the timer 345 to wait a predetermined time interval for the committed transactions to be finished (referring to Figure 3), the sequencer 440 receives one or more signals from the memory cycle tracker and command generator 450 to determine whether committed memory transactions have been completed. Furthermore, the sequencer 440 may receive one or more signals from the memory cycle tracker and command generator 450 to determine whether the memory devices have entered into a known state instead of using a timer to wait a predetermined period of time.

**[0032]** Figure 5 shows an exemplary embodiment of a computer system. The computer system 500 includes a central processing unit (CPU) 510, a memory controller (MCH) 520, a number of dual in-line memory modules (DIMMs) 525, a number of memory devices 527, an advance graphics port (AGP) 530, an input/output controller

(ICH) 540, a number of Universal Serial Bus (USB) ports 545, an audio converter co-decoder (AC Codec) 560, a switch 550, and a firmware hub 570.

**[0033]** In one embodiment, the CPU 510, the AGP 530, the DIMMS 525, and the ICH 540 are coupled to the MCH 520. The MCH 520 routes data to and from the DDR SDRAM devices 527 via the DIMMs 525. The memory devices may include various types of memories, such as, for example, dynamic random access memory (DRAM), synchronous dynamic random access memory (SDRAM), double-data rate (DDR) SDRAM, or flash memory. In one embodiment, each of the DIMMs 525 is mounted on the same motherboard (not shown) via a DIMM connector (not shown) in order to be coupled to the MCH 520. In one embodiment, the USB ports 545, the AC Codec 560, and the switch 550 are coupled to the ICH 540. The switch 550 may be further coupled to a firmware hub 570, a floppy disk drive 551, data input devices 553, such as, a keyboard, a mouse, etc., a number of serial ports 555, and a number of parallel ports 557.

**[0034]** Note that any or all of the components and the associated hardware illustrated in Figure 5 may be used in various embodiments of the computer system. However, it should be appreciated that other configuration of the computer system may include some or all of the devices shown in Figure 5. Furthermore, some embodiments of the computer system may include additional components not shown in Figure 5.

**[0035]** In one embodiment, the MCH 520 supports a deterministic shutdown mode. The MCH 520 may determine whether the memory devices 527 have been initialized by reading a memory initialization register bit, which may not be affected by the warm reset. The bit may be stored in the ICH 540. The bit may be set to “1” when the memory devices 527 are not initialized and “0” otherwise. Before the memory

devices 527 have been initialized, the MCH 520 and the supporting BIOS program causes a cold reset in the memory devices 527 if a warm reset occurs in the system 500. In one embodiment, the BIOS program triggers a power cycling sequence by sending a command to the ICH 540 to cause the ICH 540 to turn off and on the power supply to the memory devices 527 in order to cause a cold reset in the memory devices 527. After the memory devices 527 have been initialized, the MCH 520 enables the deterministic shutdown mode.

**[0036]** While in the deterministic shutdown mode, the MCH 520 gates a main system reset signal that indicates a warm reset. In response to the warm reset, the MCH 520 puts the memory devices 527 into a known state, such as a self-refresh state, such that the memory devices 527 can be recovered after the warm reset without going through a power cycling sequence. While the MCH 520 is putting the memory devices 527 into the known state, various logic blocks in the MCH 520 that are no longer used to complete putting the memory devices 527 into the known state may be reset in various stages. This is also referred to as a staged reset of the MCH 520. After the memory devices 527 have been put into the known state, the main system reset signal is passed to all logic blocks of the MCH 520 to reset the MCH 520.

**[0037]** Moreover, one should appreciate that the above technique is applicable to various types of memory devices (e.g., DRAM, SDRAM, DDR SDRAM), as well as different types of system environment, such as a multi-drop environment or a point-to-point environment.

**[0038]** The foregoing discussion merely describes some exemplary embodiments of the present invention. One skilled in the art will readily recognize from such

discussion, the accompanying drawings and the claims that various modifications can be made without departing from the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.